Short introduction to version control with git and github

Johan Nylander

Wed 11 Apr 2018

git = github?

- **git** - is a version control software
- **github** - is an online repository (github.com)

Other

- There are other version control systems (cvs, svn, . . .)
- There are other online repositories (SourceForge, Bitbucket, . . .)

Interactive tutorials

- https://try.github.io
- https://learngitbranching.js.org/

Fetch repository from github.com using git

```
git clone https://github.com/nylander/ptemplate.git
```

```
$ git clone https://github.com/nylander/ptemplate.git
Cloning into 'ptemplate'...
remote: Counting objects: 27, done.
remote: Total 27 (delta 0), reused 0 (delta 0), pack-reused
27
Unpacking objects: 100% (27/27), done.

$ ls ptemplate/
bin  data  doc  old  README.md  results  src  tmp

$
```

Version control

- Good for text and files that potentially change over time
- Great for managing going back and forth between versions of such documents
- Super for managing such files when shared by many people

git workflow

Create a folder with some files and folders

```
mkdir myproj
cd myproj
mkdir -p dir/dir2
touch apa.txt dir/bpa.txt dir/dir2/cpa.txt
```

git workflow

tree

```
$ tree
.
├── apa.txt
└── dir
    ├── bpa.txt
    └── dir2
        └── cpa.txt

2 directories, 3 files

$ 
```

git workflow

`git init`

```
$ git init
Initialized empty Git repository in /home/nylander/Documents/Projects/NRM-p
rojects/BIOinfofika/2018/GIT/myproj/.git/

$ 
```

git workflow

`ls -la`

```
$ ls -la
total 16
drwxr-xr-x 4 nylander nylander 4096 Apr 10 17:20 .
drwxr-xr-x 4 nylander nylander 4096 Apr 10 17:20 ..
-rw-r--r-- 1 nylander nylander    0 Apr 10 17:20 apa.txt
drwxr-xr-x 3 nylander nylander 4096 Apr 10 17:20 dir
drwxr-xr-x 7 nylander nylander 4096 Apr 10 17:20 .git

$
```

git workflow

`ls .git`

```
$ ls .git
branches  config  description  HEAD  hooks  info  objects  refs

$
```

git workflow

`git status`

```
$ git status
On branch master

No commits yet

Untracked files:
  (use "git add <file>..." to include in what will be committed)

        apa.txt
        dir/

nothing added to commit but untracked files present (use "git add" to track
)

$ 
```

git workflow

```
git add apa.txt dir/
```

git workflow

`git status`

```
$ git status
On branch master

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)

        new file:   apa.txt
        new file:   dir/bpa.txt
        new file:   dir/dir2/cpa.txt


$ 
```

git workflow

```
git commit -m "first commit"
```

```
$ git commit -m "first commit"
[master (root-commit) 448e0e9] first commit
 3 files changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 apa.txt
 create mode 100644 dir/bpa.txt
 create mode 100644 dir/dir2/cpa.txt

$ 
```

git workflow

`git status`

```
$ git status
On branch master
nothing to commit, working tree clean

$ 
```

git workflow

Add a readme-file in markdown format [1]

```
cat > README.md
# README for myproj

- 2018-04-01
- Joe.Bro@flo.co

## Description

Some description
```

[1] Here I use cat, but you can use any preferred method, obviously

git workflow

`git status`

```
$ git status
On branch master
Untracked files:
  (use "git add <file>..." to include in what will be committed)

        README.md

nothing added to commit but untracked files present (use "git add" to track
)

$ 
```

git workflow

```
git add README.md
git commit -m "added README.md"
```

git workflow

Edit some file

```
echo "apa" >> apa.txt
```

git workflow

`git status`

```
$    git status
On branch master
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git checkout -- <file>..." to discard changes in working directory)

        modified:   apa.txt

no changes added to commit (use "git add" and/or "git commit -a")

$
```

git workflow

```
git add apa.txt
git commit -m "edited apa.txt"
```

git workflow - Summary

```
git init
git add <files and folders>
git commit -m "message"
```

git branch

```
git branch
```

```
$ git branch
* master

$ 
```

git branch

```
git branch myfeature
git branch
```

```
$ git branch myfeature

$ git branch
* master
  myfeature

$
```

git branch

`git checkout myfeature`

```
$ git checkout myfeature
M       apa.txt
Switched to branch 'myfeature'

$ 
```

git branch

```
git rm apa.txt
ls
```

```
$ git rm apa.txt
rm 'apa.txt'

$ ls
dir  README.md

$ 
```

git branch

```
echo "foo" > bar
```

git branch

git status

```
$ git status
On branch myfeature
Changes to be committed:
  (use "git reset HEAD <file>..." to unstage)

        deleted:    apa.txt

Untracked files:
  (use "git add <file>..." to include in what will be committed)

        bar


$ 
```

git branch

```
git add bar
git commit -m "first commits on branch myfeature"
```

```
$ git add bar

$ git commit -m "first commits on branch myfeature"
[myfeature 522f40e] first commits on branch myfeature
 2 files changed, 1 insertion(+), 1 deletion(-)
 delete mode 100644 apa.txt
 create mode 100644 bar

$
```

git branch

ls

```
$ ls
bar  dir  README.md

$ 
```

git branch

```
git checkout master
```

```
$ git checkout master
Switched to branch 'master'

$ 
```

git branch

ls

```
$ ls
apa.txt  dir  README.md

$ 
```

git branch-edit-testmerge-merge workflow

```
git checkout -b myfix
sed -i 's/p/bb/' apa.txt
git add apa.txt
git commit -m "apa to abba"
git checkout master
git checkout -b mymergetest
git merge myfix # Check if all is OK
git checkout master
git merge myfix
git branch -d mymergetest
git branch -d myfix
```

Add your local project to github

Note: Requires a user account on github.com

```
cd myproject
git init
git add --all
git commit -m "first commit"
git remote add origin https://github.com/user/myproject.git
git remote -v
git push origin master
```

git undo/revert

```
echo "bpa" >> apa.txt
git add apa.txt
git commit -m "added bpa to apa.txt"
```

git undo/revert

Assume we wish to undo changes to file `apa.txt`

- Either edit the file again (then `add` + `commit`),
- or “go back” to the state before the last commit,
- or see many more options (link)

git undo/revert

```
git log
```

```
commit 9fa4000b745311ac7484bec05e255028b2027ca3 (HEAD -> master)
Author: nylander <j.a.a.nylander@gmail.com>
Date:   Tue Apr 10 18:13:48 2018 +0200

    edited apa.txt

commit b42276d27a8eeaf04b3dc324b2724a9e421d41e1
Author: nylander <j.a.a.nylander@gmail.com>
Date:   Tue Apr 10 17:58:37 2018 +0200

    edited apa.txt

commit 6e54b342e38230ae6bbd5f4d200669cdef49ec73
Author: nylander <j.a.a.nylander@gmail.com>
Date:   Tue Apr 10 17:58:24 2018 +0200

    added README

commit bb66ab165fe38151e399469bf62af2151e63d06b
Author: nylander <j.a.a.nylander@gmail.com>
:
```

git undo/revert

```
git checkout b42276d27
```